



C1 - Version: 3.1

27 April 2021

C Programming



C Programming

C1 - Version: 3.1

 **5 days Course**

Description:

The C programming language is one of the programming languages used today for software development in advanced computer environments. The use of the C programming language for software development holds many programming possibilities that one must learn to utilize, but which also demands dealing with difficulties that arise. Upon completing the course, the participant will be able to deal with real-life programming problems. The course will enrich and deepen the participant's experience in developing large and complex systems using the C programming language. This course uses many examples to explain the theoretical material, and includes many suggestions for practical use of the language

Intended audience:

This course is intended for programmers in various fields, and hardware engineers

Prerequisites:

Experience in programming in a high level language
Familiarity with data structures and operating systems - preferred

Objectives:

Upon completion of this course the participant will be able to develop software in C
Participants will learn the pros and cons of the various programming structures in C

Topics:

Introduction

- Introduction.

- History.
- Characteristics of the language.
- A simple program.
- Stages of program development.

Variables

- Units of memory
- Representation of numbers in memory
- Variable types in C
- Variables – Declaration and Usage
- Assigning values to variables
- I/O - printf(), scanf()
- Type modifiers
- Type conversion
- Overflow

Expressions and Operators

- Expressions
- Some standard library functions: getchar(), putchar()
- Random numbers
- Operators
 - Arithmetic operators
 - Assignment, increment and decrement operators
 - sizeof operator
 - Relational, logical operators

Flow Control – Conditions

- if statement
- Conditional operator – “ ? : ”
- Nested if
- switch statement
- break – continue

More Flow Control – Loops

- while
- for
- do...while

Arrays and Strings

- Arrays.
- Multidimensional arrays.
- Strings – arrays of characters.
- Library functions for manipulating strings

Functions

- Functions
 - Definition, declarations, and structure.
 - Using arguments.
 - Exiting the function.
- Passing parameters
 - By value.
 - By reference.
- Scope and visibility
- Lifetime of variables

Pointers

- Why pointers?
- Pointers – declaration, assignment, display
- Pointer pitfalls
- Pointer and arrays
- Arithmetic operations on pointers
- Comparison of pointers
- Scanning arrays using pointers
- Pointers and functions

- Passing variables by reference

Dynamic Memory Allocation

- Dynamic Memory Allocation – why?
- malloc(), calloc(), free() library functions
- Dynamic Memory Allocation for o-dimensional arrays
- Arrays of pointers
- Dynamic memory allocation for two-dimensional arrays

Structs and Unions

- typedef
- enum
- Structure
 - Definition.
 - Definition of variables.
 - Referencing structures.
- Array of Structures
 - Usage.
 - Initialization.
- Nested Structures.
- Unions

Bits

- Bitwise operators: AND, OR, XOR, NOT
- Left shift, right shift